Logo

STUDENT REPORT

DETAILS

Name

ZIHAN

Roll Number

3BR23EE113

EXPERIMENT

Title

OBJECT SCORE

Description

In a family, there are N members each have a capacity of Ci units to buy anything. In a store there are M objects. Each of which have some price Pi and weight Wi print on it. Each of the members go to the store and can buy all those items whose price is less than or equal to their buying capacity and store that bought object in a bag. Find the maximum weight of each of the bags collected by all N members individually.

Input Format:
First line contains two integers N and M where N is the number of members in the house and M is the number of objects in the store.
Second line contains N space-separated integers (C1, C2, C3,...)

the next M lines contains each object price and weight(Pi,Wi) as space seperated integers.

Sample Input:

3 4

10 20 30

5 10

15 20

10 25

20 30

Sample Output:
35 85 85

Source Code:

```
def maximum_weights(N, M, capacities, objects):
    # Sort objects by price
    objects.sort(key=lambda x: x[0])

    results = []

    for capacity in capacities:
        max_weight = 0
        # Calculate maximum weight for the current mem
ber
        for price, weight in objects:
            if price <= capacity:
                max_weight += weight
            else:
                break  # No need to check further as t
he list is sorted by price
        results.append(max_weight)

    return results

# Input handling
N, M = map(int, input().strip().split())  # Read N and
M
capacities = list(map(int, input().strip().split()))
# Read capacities
objects = [tuple(map(int, input().strip().split())) fo
r _ in range(M)]  # Read objects

# Get results
result = maximum_weights(N, M, capacities, objects)

# Print results
print(" ".join(map(str, result)))
```

RESULT

2 / 5 Test Cases Passed | 40 %